

LAMPIRAN 1

DATA JARAK DAN WAKTU ANTAR LOKASI

	Pulomas	Lebak Bulus	Jatinegara	Tanjung Priok	Tanah Abang	Kebon Jeruk	Kalideres	Pluit	Cililitan	Pasar Minggu	Pasar Senen
	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))	(Waktu(1,2,3,4))
Pulomas	(0,0,0,0)	(60,63,65,66)	(14,15,18,20)	(23,25,30,32)	(34,35,37,40)	(50,51,52,54)	(64,66,70,73)	(45,46,48,50)	(23,25,29,30)	(42,45,47,51)	(17,20,24,25)
Lebak Bulus	(60,63,65,66)	(0,0,0,0)	(45,50,55,61)	(57,60,70,72)	(41,44,45,50)	(44,46,49,50)	(55,56,60,62)	(56,58,60,64)	(37,38,40,45)	(19,20,22,25)	(55,60,63,65)
Jatinegara	(14,15,18,20)	(45,50,55,61)	(0,0,0,0)	(37,38,40,42)	(26,31,36,41)	(60,65,70,75)	(94,99,102,107)	(63,67,70,74)	(23,25,26,28)	(40,45,50,55)	(22,27,30,35)
Tanjung Priok	(23,25,30,32)	(57,60,70,72)	(37,38,40,42)	(0,0,0,0)	(38,43,46,48)	(63,68,71,75)	(84,89,94,99)	(38,43,48,53)	(47,50,53,56)	(69,71,73,75)	(30,35,40,45)
Tanah Abang	(34,35,37,40)	(41,44,45,50)	(26,31,36,41)	(38,43,46,48)	(0,0,0,0)	(34,36,37,40)	(54,59,64,69)	(40,45,50,55)	(49,54,59,64)	(39,42,45,48)	(27,29,32,34)
Kebon Jeruk	(50,51,52,54)	(44,46,49,50)	(60,65,70,75)	(63,68,71,75)	(34,36,37,40)	(0,0,0,0)	(43,47,50,52)	(44,49,54,59)	(52,56,59,63)	(49,53,57,61)	(44,49,54,59)
Kalideres	(64,66,70,73)	(55,56,60,62)	(94,99,102,107)	(84,89,94,99)	(54,59,64,69)	(43,47,50,52)	(0,0,0,0)	(44,49,54,59)	(74,76,80,81)	(76,79,82,85)	(72,74,76,78)
Pluit	(45,46,48,50)	(56,58,60,64)	(63,67,70,74)	(38,43,48,53)	(40,45,50,55)	(44,49,54,59)	(44,49,54,59)	(0,0,0,0)	(47,50,53,56)	(46,51,56,61)	(50,52,54,56)
Cililitan	(23,25,29,30)	(37,38,40,45)	(23,25,26,28)	(47,50,53,56)	(49,54,59,64)	(52,56,59,63)	(74,76,80,81)	(47,50,53,56)	(0,0,0,0)	(27,32,37,42)	(35,40,45,50)
Pasar Minggu	(42,45,47,51)	(19,20,22,25)	(40,45,50,55)	(69,71,73,75)	(39,42,45,48)	(49,53,57,61)	(76,79,82,85)	(46,51,56,61)	(27,32,37,42)	(0,0,0,0)	(51,53,55,57)
Pasar Senen	(17,20,24,25)	(55,60,63,65)	(22,27,30,35)	(30,35,40,45)	(27,29,32,34)	(44,49,54,59)	(72,74,76,78)	(50,52,54,56)	(35,40,45,50)	(51,53,55,57)	(0,0,0,0)

Gambar 5.1: Data pengamatan 4 waktu tempo

	Pulomas		Lebak Bulus		Jatinegara		Tanjung Priok		Tanah Abang		Kebon Jeruk		Kalideres		Pluit		Cililitan		Pasar Minggu		Pasar Senen	
	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak	Waktu	Jarak
Pulomas	0 m	0 km	63,42 m	24 km	16,78 m	6 km	27,5 m	11 km	36,58 m	10 km	51,8 m	19 km	68,28 m	27 km	47,28 m	22 km	26,72 m	9 km	46,3 m	18 km	21,44 m	5 km
Lebak Bulus	63,42 m	24 km	0 m	0 km	52,79 m	23 km	64,73 m	37 km	45,13 m	16 km	47,22 m	21 km	58,27 m	26 km	59,6 m	27 km	40,2 m	17 km	21,58 m	7,8 km	60,61 m	30 km
Jatinegara	16,78 m	6 km	52,79 m	23 km	0 m	0 km	39,28 m	17 km	33,5 m	12 km	67,5 m	22 km	100,5 m	33 km	68,5 m	26 km	25,5 m	5,8 km	47,5 m	18 km	28,5 m	12 km
Tanjung Priok	27,5 m	11 km	64,73 m	37 km	39,28 m	17 km	0 m	0 km	43,61 m	21 km	69,2 m	20 km	91,5 m	28 km	45,5 m	19 km	51,5 m	21 km	72 m	33 km	37,5 m	11 km
Tanah Abang	36,58 m	10 km	45,13 m	16 km	33,5 m	12 km	43,61 m	21 km	0 m	0 km	36,81 m	11 km	61,5 m	19 km	47,5 m	15 km	56,5 m	13 km	43,5 m	15 km	30,5 m	7,3 km
Kebon Jeruk	51,8 m	19 km	47,22 m	21 km	67,5 m	22 km	69,2 m	20 km	36,81 m	11 km	0 m	0 km	47,92 m	14 km	51,5 m	16 km	57,5 m	21 km	55 m	22 km	51,5 m	19 km
Kalideres	68,28 m	27 km	58,27 m	26 km	100,5 m	33 km	91,5 m	28 km	61,5 m	19 km	47,92 m	14 km	0 m	0 km	51,5 m	16 km	77,73 m	35 km	82,17 m	31 km	75 m	28 km
Pluit	47,28 m	22 km	59,6 m	27 km	68,5 m	26 km	45,5 m	19 km	47,5 m	15 km	51,5 m	16 km	51,5 m	16 km	0 m	0 km	51,5 m	27 km	53,5 m	26 km	53 m	20 km
Cililitan	26,72 m	9 km	40,2 m	17 km	25,5 m	5,8 km	51,5 m	21 km	56,5 m	13 km	57,5 m	21 km	77,73 m	35 km	51,5 m	27 km	0 m	0 km	34,5 m	9,1 km	42,5 m	13 km
Pasar Minggu	46,3 m	18 km	21,58 m	7,8 km	47,5 m	18 km	72 m	33 km	43,5 m	15 km	55 m	22 km	82,17 m	31 km	53,5 m	26 km	34,5 m	9,1 km	0 m	0 km	54 m	24 km
Pasar Senen	21,44 m	5 km	60,61 m	30 km	28,5 m	12 km	37,5 m	11 km	30,5 m	7,3 km	51,5 m	19 km	75 m	28 km	53 m	20 km	42,5 m	13 km	54 m	24 km	0 m	0 km

Gambar 5.2: Data pengamatan waktu tempo yang telah di defuzzifikasi dan jarak antar lokasi

LAMPIRAN 2

PERHITUNGAN NILAI PENALTY

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 1	08.00	63,42	09.04	09.00	09.04	8	09.19	17.00	0		
1 ke 4	09.19	45,13	10.04	09.00	10.04	0	10.19	17.00	0		
4 ke 2	10.19	33,5	11.04	09.00	10.53	0	11.08	17.00	0		
2 ke 10	11.08	28,5	11.37	09.00	11.37	0	11.52	17.00	0		
10 ke 7	11.52	53	12.45	09.00	12.45	0	13.00	17.00	0		
7 ke 6	13.00	51,5	13.52	09.00	13.52	0	14.07	17.00	0		
6 ke 9	14.07	82,17	15.29	09.00	15.29	0	15.44	17.00	0		
9 ke 8	15.44	34,5	16.19	09.00	16.19	0	16.34	17.00	0		
8 ke 3	16.34	51,5	17.26	09.00	17.26	0	17.41	17.00	41		
3 ke 5	17.41	69,2	18.50	09.00	18.50	0	19.05	17.00	125		
5 ke 0	19.05	51,8	19.57	09.00	19.57	0	20.12	17.00	192		
									358	Total Penalti	

Gambar 5.3: Perhitungan Nilai Penalty Kromosom 1

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 8	08.00	26,72	08.27	09.00	09.00	33	09.15	17.00	0		
8 ke 9	09.15	34,5	09.50	09.00	09.50	0	10.05	17.00	0		
9 ke 2	10.05	47,5	10.53	09.00	10.53	0	11.08	17.00	0		
2 ke 6	11.08	100,5	12.49	09.00	12.49	0	13.04	17.00	0		
6 ke 7	13.04	51,5	13.56	09.00	13.56	0	14.11	17.00	0		
7 ke 1	14.11	59,6	15.11	09.00	15.11	0	15.26	17.00	0		
1 ke 5	15.26	47,22	16.13	09.00	16.13	0	16.28	17.00	0		
5 ke 10	16.28	51,5	17.20	09.00	17.20	0	17.35	17.00	35		
10 ke 4	17.35	30,5	18.06	09.00	18.06	0	18.21	17.00	81		
4 ke 3	18.21	43,61	19.05	09.00	19.05	0	19.20	17.00	140		
3 ke 0	19.20	27,5	19.48	09.00	19.48	0	20.03	17.00	183		
									439	Total Penalti	

Gambar 5.4: Perhitungan Nilai Penalty Kromosom 2

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 7	08.00	47,28	08.47	09.00	09.00	13	09.15	17.00	0		
7 ke 6	09.15	51,5	10.07	09.00	10.07	0	10.22	17.00	0		
6 ke 5	10.22	47,92	11.10	09.00	11.10	0	11.25	17.00	0		
5 ke 1	11.25	47,22	12.12	09.00	12.12	0	12.27	17.00	0		
1 ke 8	12.27	40,2	13.07	09.00	13.07	0	13.22	17.00	0		
8 ke 2	13.22	25,5	13.48	09.00	13.48	0	14.03	17.00	0		
2 ke 4	14.03	33,5	14.37	09.00	14.37	0	14.52	17.00	0		
4 ke 9	14.52	43,5	15.36	09.00	15.36	0	15.51	17.00	0		
9 ke 10	15.51	54	16.45	09.00	16.45	0	17.00	17.00	0		
10 ke 3	17.00	37,5	17.38	09.00	17.38	0	17.53	17.00	53		
3 ke 0	17.53	27,5	18.21	09.00	18.21	0	18.36	17.00	96		
									149	Total Penalti	

Gambar 5.5: Perhitungan Nilai Penalty Kromosom 3

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 10	08.00	21,44	08.21	09.00	09.00	39	09.15	17.00	0		
10 ke 9	09.15	54	10.09	09.00	10.09	0	10.24	17.00	0		
9 ke 6	10.24	82,17	11.46	09.00	11.46	0	12.01	17.00	0		
6 ke 3	12.01	91,5	13.33	09.00	13.33	0	13.48	17.00	0		
3 ke 8	13.48	51,5	14.40	09.00	14.40	0	14.55	17.00	0		
8 ke 4	14.55	56,5	15.52	09.00	15.52	0	16.07	17.00	0		
4 ke 2	16.07	33,5	16.41	09.00	16.41	0	16.56	17.00	0		
2 ke 1	16.56	52,79	17.49	09.00	17.49	0	18.04	17.00	64		
1 ke 5	18.04	47,22	18.51	09.00	18.51	0	19.06	17.00	126		
5 ke 7	19.06	51,5	19.58	09.00	19.58	0	20.13	17.00	193		
7 ke 0	20.13	47,28	21.00	09.00	21.00	0	21.15	17.00	255		
									638	Total Penalti	

Gambar 5.6: Perhitungan Nilai Penalty Kromosom 4

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 6	08.00	68,28	09.08	09.00	09.08	0	09.23	17.00	0		
6 ke 9	09.23	82,17	10.45	09.00	10.45	0	11.00	17.00	0		
9 ke 8	11.00	34,5	11.35	09.00	11.35	0	11.50	17.00	0		
8 ke 1	11.50	40,2	12.32	09.00	12.32	0	12.45	17.00	0		
1 ke 10	12.48	60,61	13.46	09.00	13.46	0	14.01	17.00	0		
10 ke 2	14.01	28,5	14.30	09.00	14.30	0	14.45	17.00	0		
2 ke 7	14.45	68,5	15.54	09.00	15.54	0	16.09	17.00	0		
7 ke 5	16.09	51,5	17.01	09.00	17.01	0	17.16	17.00	16		
5 ke 3	17.16	69,2	18.26	09.00	18.26	0	18.41	17.00	101		
3 ke 4	18.41	43,61	19.25	09.00	19.25	0	19.40	17.00	160		
4 ke 0	19.40	36,58	20.27	09.00	20.27	0	20.42	17.00	222		
									499	Total Penalti	

Gambar 5.7: Perhitungan Nilai Penalty Kromosom 5

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 10	08.00	21,44	08.21	09.00	09.00	39	09.15	17.00	0		
10 ke 5	09.15	51,5	10.07	09.00	10.07	0	10.22	17.00	0		
5 ke 2	10.22	67,5	11.30	09.00	11.30	0	11.45	17.00	0		
2 ke 8	11.45	25,5	12.11	09.00	12.11	0	12.26	17.00	0		
8 ke 4	12.26	56,5	13.23	09.00	13.23	0	13.38	17.00	0		
4 ke 1	13.38	45,13	14.24	09.00	14.24	0	14.39	17.00	0		
1 ke 7	14.39	59,6	15.39	09.00	15.39	0	15.54	17.00	0		
7 ke 6	15.54	51,5	16.46	09.00	16.46	0	17.01	17.00	1		
6 ke 9	17.01	82,17	18.23	09.00	18.23	0	18.38	17.00	98		
9 ke 3	18.38	72	19.50	09.00	19.50	0	20.05	17.00	185		
3 ke 0	20.05	27,5	20.23	09.00	20.23	0	20.48	17.00	228		
									512	Total Penalti	

Gambar 5.8: Perhitungan Nilai Penalty Kromosom 6

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 5	08.00	51,8	08.52	09.00	09.00	8	09.15	17.00	0		
5 ke 6	09.15	47,92	10.03	09.00	10.03	0	10.18	17.00	0		
6 ke 4	10.18	61,5	11.20	09.00	11.20	0	11.35	17.00	0		
4 ke 2	11.35	33,5	12.09	09.00	12.09	0	12.24	17.00	0		
2 ke 3	12.24	39,28	13.03	09.00	13.03	0	13.18	17.00	0		
3 ke 7	13.18	45,5	14.04	09.00	14.04	0	14.19	17.00	0		
7 ke 10	14.19	53	15.12	09.00	15.12	0	15.27	17.00	0		
10 ke 9	15.27	54	16.21	09.00	16.21	0	16.36	17.00	0		
9 ke 8	16.36	34,5	17.11	09.00	17.11	0	17.26	17.00	26		
8 ke 1	17.26	40,2	18.06	09.00	18.06	0	18.21	17.00	81		
1 ke 0	18.21	63,42	19.24	09.00	19.24	0	19.39	17.00	159		
									266	Total Penalti	

Gambar 5.9: Perhitungan Nilai Penalty Kromosom 7

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 9	08.00	46,3	08.46	09.00	09.00	14	09.15	17.00	0		
9 ke 5	09.15	55	10.10	09.00	10.10	0	10.25	17.00	0		
5 ke 6	10.25	47,92	11.13	09.00	11.13	0	11.28	17.00	0		
6 ke 8	11.28	77,73	12.46	09.00	12.46	0	13.01	17.00	0		
8 ke 2	13.01	25,5	13.27	09.00	13.27	0	13.42	17.00	0		
2 ke 4	13.42	33,5	14.16	09.00	14.16	0	14.31	17.00	0		
4 ke 10	14.31	30,5	15.02	09.00	15.02	0	15.17	17.00	0		
10 ke 3	15.17	37,5	15.55	09.00	15.55	0	16.10	17.00	0		
3 ke 7	16.10	45,5	16.56	09.00	16.56	0	17.11	17.00	11		
7 ke 1	17.11	59,6	18.11	09.00	18.11	0	18.26	17.00	86		
1 ke 0	18.26	63,42	19.29	09.00	19.29	0	19.44	17.00	164		
									261	Total Penalti	

Gambar 5.10: Perhitungan Nilai Penalty Kromosom 8

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 9	08.00	21,44	08.21	09.00	09.00	39	09.15	17.00	0		
9 ke 8	09.15	34,5	09.50	09.00	09.50	0	10.05	17.00	0		
8 ke 1	10.05	40,2	10.45	09.00	10.45	0	11.00	17.00	0		
1 ke 2	11.00	52,79	11.53	09.00	11.53	0	12.08	17.00	0		
2 ke 6	12.08	100,5	13.49	09.00	13.49	0	14.04	17.00	0		
6 ke 10	14.04	75	15.19	09.00	15.19	0	15.34	17.00	0		
10 ke 7	15.34	53	16.27	09.00	16.27	0	16.42	17.00	0		
7 ke 4	16.42	47,5	17.30	09.00	17.30	0	17.45	17.00	45		
4 ke 3	17.45	43,61	18.29	09.00	18.29	0	18.44	17.00	104		
3 ke 5	18.44	69,2	19.53	09.00	19.53	0	20.08	17.00	188		
5 ke 0	20.08	51,8	21.00	09.00	21.00	0	21.15	17.00	255		
									592	Total Penalti	

Gambar 5.11: Perhitungan Nilai Penalty Kromosom 9

Node	Start Time	Lama Jalan	Tiba	Earliest Time	Mulai	Tunggu	Selesai	Latest Time	Penalti		
0 ke 6	08.00	68,28	09.08	09.00	09.08	0	09.23	17.00	0		
6 ke 2	09.23	100,5	11.04	09.00	11.04	0	11.19	17.00	0		
2 ke 10	11.19	28,5	11.48	09.00	11.48	0	12.03	17.00	0		
10 ke 8	12.03	42,5	12.46	09.00	12.46	0	13.01	17.00	0		
8 ke 3	13.01	51,5	13.53	09.00	13.53	0	14.08	17.00	0		
3 ke 5	14.08	69,2	15.17	09.00	15.17	0	15.32	17.00	0		
5 ke 7	15.32	51,5	16.24	09.00	16.24	0	16.39	17.00	0		
7 ke 4	16.39	47,5	17.27	09.00	17.27	0	17.42	17.00	42		
4 ke 9	17.42	43,5	18.26	09.00	18.26	0	18.41	17.00	101		
9 ke 1	18.41	21,58	19.03	09.00	19.03	0	19.18	17.00	138		
1 ke 0	19.18	63,42	20.21	09.00	20.21	0	20.36	17.00	216		
									497	Total Penalti	

Gambar 5.12: Perhitungan Nilai Penalty Kromosom 10

LAMPIRAN 3

KODINGAN PROGRAM

```
#Memuat File
import numpy as np
from random import randint as rni
from random import random as rnf
import pandas as pd
from csv import reader
from itertools import permutations as permut

#Menyiapkan Data
theFile = open('JarakSpasial.csv','r')
JarakSpasial = reader(theFile)
JarakSpasial = list(JarakSpasial)
theFile.close()
spasial = np.zeros((11,11),dtype=float)

theFile = open('JarakTemporal.csv','r')
JarakTemporal = reader(theFile)
JarakTemporal = list(JarakTemporal)
theFile.close()
temporal = np.zeros((11,11),dtype=float)

# generating matrix for both data
for i in range(len(JarakSpasial)):
    for j in range(len(JarakSpasial)):
        spas = float(JarakSpasial[i][j])
        temp = float(JarakTemporal[i][j])
        spasial[i,j] = spas
        temporal[i,j] = temp

agen = ['Pulomas','Lebak Bulus','Jatinegara',
        'Tanjung Priuk','Tanah Abang','Kebon Jeruk',
        'Kalideres','Pluit','Cililitan','Pasar Minggu','Pasar Senen']
```

```

dfSpas = pd.DataFrame(spasial, columns = agen, index = agen)
dfTemp = pd.DataFrame(temporal, columns = agen, index = agen)

#Inisialisasi Populasi
def randIntSquMat(n):
M = []
for loop in range(n):
array = np.zeros(n,dtype=int)
for i in range(len(array)):
randomValue = rni(1,n)
while randomValue in array:
randomValue = rni(1,n)
array[i] = randomValue
M.append(array)
M = np.asarray(M)
return(M)
populasi1 = randIntSquMat(10)
print(populasi1)

#Total Jarak Spasial dan Waktu
def jarakTempuh(K): # jarak yang ditempuh dari rute pada suatu kromosom
jalur = [0]
for dna in K:
jalur.append(dna)
jalur.append(0)
total = 0
for i in range(len(jalur)-1):
part = [jalur[i],jalur[i+1]]
total += spasial[part[0],part[1]]
return(total)

def waktuTempuh(K): # waktu yang ditempuh dari rute pada suatu kromosom
jalur = [0]
for dna in K:
jalur.append(dna)
jalur.append(0)

```

```

total = 0
for i in range(len(jalur)-1):
    part = [jalur[i],jalur[i+1]]
    total += temporal[part[0],part[1]]
return(total)

listTotalSpasial = []
listTotalTemporal = []
for kromosom in populasi1:
    listTotalSpasial.append(jarakTempuh(kromosom))
    listTotalTemporal.append(waktuTempuh(kromosom))
print('Jarak tempuh setiap kromosom:')
print(np.asarray(listTotalSpasial))
print('Waktu tempuh setiap kromosom:')
print(np.asarray(listTotalTemporal))

#Menghitung Fitness
listRasioSpasial = []
for kromosom in listTotalSpasial:
    rasio = kromosom/sum(listTotalSpasial)
    listRasioSpasial.append(rasio)
listRasioTemporal = []
for kromosom in listTotalTemporal:
    rasio = kromosom/sum(listTotalTemporal)
    listRasioTemporal.append(rasio)
listNilaiFitness = []
for index in range(len(listTotalSpasial)):
    penyebut = listRasioSpasial[index]+listRasioTemporal[index]
    fitness = 1/penyebut
    listNilaiFitness.append(fitness)
totalFitness = sum(listNilaiFitness)
listRasioFitness = []
for fitness in listNilaiFitness:
    rasio = fitness/totalFitness
    listRasioFitness.append(rasio)
kumulatifFitness = [listRasioFitness[0]]

```

```

current = listRasioFitness[0]
for index in range(1,len(listRasioFitness)):
    current += listRasioFitness[index]
    kumulatifFitness.append(current)
for i in kumulatifFitness:
    print(i)

#Seleksi
# listRN isi bilangan riil acak (0,1), yang akan dicek ada di
interval kumulatif yang mana
listRN = []
for i in range(10):
    listRN.append(rnf())

listKelasRN = []
for value in listRN:
    for index in range(len(kumulatifFitness)):
        if value < kumulatifFitness[index]:
            listKelasRN.append(index)
            break

for index in range(len(kumulatifFitness)):
    print(kumulatifFitness[index])
    print('\n')
for index in range(len(listRN)):
    print(listRN[index], '--->', listKelasRN[index])

#populasi2: kromosomnya udah ditiban dengan yang baru; jadi ada
yg ga unik
populasi2 = []
for i in range(len(populasi1)):
    x = listKelasRN[i]
    populasi2.append(populasi1[x])
    print('populasi1 (sebelum seleksi):')
for i in range(len(populasi1)):
    print(populasi1[i], 'tukar dengan', populasi1[listKelasRN[i]],

```



```

'(index ke {})'.format(listKelasRN[i]))
print('populasi2 (setelah seleksi):')
for i in range(len(populasi2)):
    print(populasi2[i])

#Rekombinasi
pr = 0.3
# acakRecom correspond dengan populasi2 (hasil seleksi)
acakRekom = []
for i in range(10):
    acakRekom.append(rnf())

kurangDari = []
for i in range(len(acakRekom)):
    if acakRekom[i] < pr:
        kurangDari.append(i+1)
print('Urutan kromosom yang kurang dari pr:',kurangDari)
disilangkan = []
for i in kurangDari:
    disilangkan.append(populasi2[i-1])
print('Kromosom yang akan disilangkan:')
for silang in disilangkan:
    print(silang)

def toLine(A,B):
    if (type(A) == type(B)) and (len(A) == len(B)) and (type(A) == type([])):
        P = []
        for i in range(len(A)):
            P.append(B[len(A)-1-i])
            P.append(A[len(A)-1-i])
        return(P)
    else: return(None)
def deployLine(L):
    A = []
    B = []
    for i in range(len(L)):

```

```

if i%2 == 0:
    A.append(L[len(L)-1-i])
    B.append(L[len(L)-2-i])
pair = [A,B]
return(pair)
def salin(L):
    S = []
    try:
        for element in L:
            S.append(element)
        return(S)
    except:
        return(None)
def uniqueIn(v,L): # v (a certain value), L (a list)
    count = 0
    for i in range(len(L)):
        if L[i] == v:
            count += 1
    if count == 1:
        return(True)
    else:
        return(False)
def kawin(A,B,t): # A (induk I), B (induk II), t (titik potong);
    t mulai dari 1
    silang = []
    for i in range(len(A)):
        if i < (t-1):
            silang.append(A[i])
        else:
            silang.append(B[i])
    return(silang)

# titik potong
tiPot = rni(1,10)
print(tiPot)

```

```

# Hasil persilangan
listHasilSilang = []
indexPart = [0] # daftar index untuk memartisi
ind = 0
if len(disilangkan) > 1:
    for i in range(len(disilangkan)):
        for j in range(len(disilangkan)):
            if i != j:
                hasilSilang = kawin(disilangkan[i],disilangkan[j],tiPot)
                listHasilSilang.append(hasilSilang)
            ind += 1
        indexPart.append(ind)
    print('Indeks Partisi:',indexPart)
    for hasil in listHasilSilang:
        print(hasil)

# pairs: pasangan antara induk dan anak
# indexPart yang tadi dipakai tuh untuk mengelompokkan antara induk
dan anak; tiap induk kan ada (n-1) anak
pairs = []
for i in range(len(disilangkan)):
    induk = disilangkan[i]
    indexAwal = range(indexPart[i],indexPart[i+1])
    for ind in indexAwal:
        pair = np.zeros((2,10),dtype=int)
        pair = [induk,listHasilSilang[ind]]
        pairs.append(pair)
    for pair in pairs:
        print(np.asarray(pair),'\n')

# pair adalah list of lists; 2 elemen: induk dan hasil silang mentah.
def rekombinasil(pair):
    induk = list(pair[0])
    anak = list(pair[1])
    line = toLine(induk,anak)

```

```

passed = [] # step yang diambil di bawah
for i in range(len(line)):
    if i%2 == 0:
        X = salin(induk)
        Y = salin(anak)
        X[induk.index(line[i])] = -216
        imaLine = toLine(X,Y)
        keyDex = imaLine.index(-216)
        current = line[i]
        path = []
        for j in range(i,(len(line)-1)):
            if (i > keyDex) and (i == j) and (line[j] not in passed):
                # sama aja kayak step not in passed
                passed.append(line[j])
                break
            elif i > keyDex:
                break
            if len(path) == 0:
                step = line[j]
                path.append(step)
                if (i == j) and (step not in passed):
                    passed.append(step)
            elif (j%2 == 1) and (line[j-1] == step) and (line[j] == current):
                # non-niban
                step = line[j]
                path.append(step)
            elif (j%2 == 0) and (line[j] == step): # non-niban
                step = line[j]
                path.append(step)
            if (i == j) and (step not in passed):
                passed.append(step)
            elif (j%2 == 1) and (line[j-1] == step): # non-niban
                step = line[j]
                path.append(step)
            elif (j > 1) and (j%2 == 0) and (line[j] != step) and
                (line[j+1] == current):

```

```

step = line[j]
path.append(step)
if (i == j) and (step not in passed):
    passed.append(step)
deployed = deployLine(line)
missed = []
doubled = []
for a in deployed[0]:
    if a not in deployed[1]:
        missed.append(a)
for i in range(len(deployed[0])):
    count = 0
    for j in range(len(deployed[1])):
        if deployed[0][i] == deployed[1][j]:
            count += 1
    if (count == 2):
        doubled.append(deployed[0][i])
return(deployed,passed,missed,doubled)

def rekombinasi2(pair):
    Rek1 = rekombinasi1(pair)
    Rdep = Rek1[0] # 2 lists in list
    Rpass = Rek1[1]
    Rmiss = Rek1[2]
    Rdou = Rek1[3]
    if (len(Rdep[0]) == len(Rpass)) and (len(Rmiss) == 0) and
        (len(Rdou) == 0):
        return(Rdep,Rpass,Rmiss,Rdou)
    else:
        R1 = Rdep[0] # induk
        R2 = Rdep[1] # solusi
        if (len(Rdou) == 1) and (len(Rmiss) == 1):
            theMissing = Rmiss[0]
            theDoubled = Rdou[0]
            indexOfDoubled = R2.index(theDoubled)
            R2[indexOfDoubled] = theMissing

```

```

result = [R1,R2]
return(result,Rpass,Rmiss,Rdou)
else:
result = [R1,R2]
return(result,Rpass,Rmiss,Rdou)

# rekombinasi3 ()
def rekombinasi3(pair): # inputannya adalah output dari
rekombinasi2() yang juga output dari rekombinasi1()
Rek2 = rekombinasi2(pair)
Rdep = Rek2[0]
Rpass = Rek2[1]
Rmiss = Rek2[2]
Rdou = Rek2[3]
if (len(Rdep[0]) == len(Rpass)) and (len(Rmiss) == 0) and
(len(Rdou) == 0):
return(Rdep,Rpass,Rmiss,Rdou)
elif (len(Rdou) == len(Rmiss)):
induk = Rdep[0]
anak = Rdep[1]
indAnak = []
# index di anak yang akan ditiban; yaitu elemen yg kedobelan
listPaketSwitch = []
for i in range(len(Rdou)):
for j in range(len(anak)):
if anak[j] == Rdou[i]:
indAnak.append(j)
break
indMissed = list(range(len(Rmiss)))
permutMissed = permut(indMissed)
for per in permutMissed:
paketSwitch = [indAnak,list(per)]
listPaketSwitch.append(paketSwitch)
listAnak = [] # list calon anak baru
for paketSwitch in listPaketSwitch:
Sanak = salin(anak)

```

```

for i in range(len(paketSwitch[0])):
    Sanak[paketSwitch[0][i]] = Rmiss[paketSwitch[1][i]]
    listAnak.append(Sanak)
for sanak in listAnak:
    R = rekombinasi2([induk,sanak])
    if (len(R[2]) == 0) and (len(R[3]) == 0):
        return(R[0])

def rekombinasi(pair):
    Rek1 = rekombinasi1(pair) # rekombinasi1
    Rdep = Rek1[0]
    Rpass = Rek1[1]
    Rmiss = Rek1[2]
    Rdou = Rek1[3]
    if (len(Rpass) == len(Rdep[0])) and (len(Rmiss) == 0) and
        (len(Rdou) == 0):
        return(Rdep)
    elif (len(Rdou) == 1) and (len(Rmiss) == 1): # rekombinasi2
        Rek2 = rekombinasi2(pair)
        Rdep = Rek2[0]
        Rpass = Rek2[1]
        Rmiss = Rek2[2]
        Rdou = Rek2[3]
        return(Rdep)
    elif (len(Rdou) == len(Rmiss)): # rekombinasi3
        Rek3 = rekombinasi3(pair)
        return(Rek3)

hasilRekombinasi = []
for i in range(len(pairs)):
    print('Sebelum rekombinasi')
    p = np.asarray(pairs[i])
    print(p)
    print('Setelah rekombinasi')
    pRekom = rekombinasi(p)
    hasilRekombinasi.append(pRekom[1])

```

```

print(np.asarray(pRekom))
print('\n')

print('Indeks Partisi (pengelompokkan hasil rekombinasi
berdasarkan induk):',indexPart)
indRekom = []
for i in range(len(indexPart)-1):
    kelas = range(indexPart[i],indexPart[i+1])
    kumpulanJarak = []
    for j in kelas:
        jarak = jarakTempuh(hasilRekombinasi[j])
        kumpulanJarak.append(jarak)
        jarakMin = min(kumpulanJarak)
    for j in kelas:
        jarak = jarakTempuh(hasilRekombinasi[j])
        if jarak == jarakMin:
            indRekom.append(j)
    break
for i in range(len(indexPart)-1):
    kelas = range(indexPart[i],indexPart[i+1])
    for j in kelas:
        print(hasilRekombinasi[j],'jarak tempuh:'
        ,jarakTempuh(hasilRekombinasi[j]))
    print('\n')
print('Indeks kromosom hasil rekombinasi dengan jarak spasial
terpendek dari setiap kelas:',indRekom)

print('Urutan (bukan index) kromosom dari populasi2 yang kurang
dari pr:',kurangDari)

populasi3 = salin(populasi2)
for i in range(len(kurangDari)):
    populasi3[kurangDari[i]-1] = hasilRekombinasi[indRekom[i]]
print('populasi2:')
walk = 0
for i in range(len(populasi2)):

```



```

if i == (kurangDari[walk]-1):
    print(populasi2[i], 'diganti dengan', hasilRekombinasi[indRekom[walk]])
    if len(kurangDari)-1 > walk:
        walk += 1
    else:
        print(populasi2[i])
        print('populasi3:')
        for i in range(len(populasi3)):
            print(populasi3[i])

#Mutasi
pm = 0.1
listRandMutasi = []
for i in range(10):
    r = rni(1,99)
    while r in listRandMutasi:
        r = rni(1,99)
    listRandMutasi.append(r)
print(listRandMutasi)

def mutasi(L,P): # L adalah listRandMutasi, P adalah populasi
    (terbaru) yang akan dimutasi
    for m in L:
        if m%10 == 0:
            baris = (m//10)
            #kolom = m-(m//10)*10
            (P[baris-1][len(P[baris-1])-1], P[baris-1][0]) =
                (P[baris-1][0], P[baris-1][len(P[baris-1])-1])
        else:
            baris = (m//10)+1
            kolom = m-(m//10)*10
            (P[baris-1][kolom-1], P[baris-1][kolom]) =
                (P[baris-1][kolom], P[baris-1][kolom-1])
    return(P)

```

```

print('Urutan dari elemen yang akan dimuasi',listRandMutasi,'\n')
print('populasi3 (hasil rekombinasi):')
print(np.asarray(populasi3))
print('populasi4 (hasil mutasi):')
populasi4 = mutasi(listRandMutasi,populasi3)
print(np.asarray(populasi4))

#Solusi Akhir
daftarJarak = []
for i in range(len(populasi4)):
    print(populasi4[i],'- jarak:',jarakTempuh(populasi4[i]))
    daftarJarak.append(jarakTempuh(populasi4[i]))
for i in range(len(populasi4)):
    if jarakTempuh(populasi4[i]) == min(daftarJarak):
        print('Kromosom yang dipilih: Z{}'.format(i+1))

#Rute Terbaik
print('Maka rute yang akan dilalui adalah:')
for i in range(len(pilih)):
    L = [agen[0]]
    for lokasi in pilih[i]:
        L.append(agen[lokasi])
    L.append(agen[0])
    S = str()
    for j in range(len(L)):
        S += L[j]
    if j != (len(L)-1):
        S += ' - '
    print(S)
    if i != (len(pilih)-1):
        print('atau')

```

DAFTAR RIWAYAT HIDUP



MUHAMMAD ABDUR ROZAQ. Lahir di Tangerang, 31 Januari 1998. Anak terakhir dari pasangan Bapak Abdul Wahid dan Ibu Encum Sumarmi. Saat ini bertempat tinggal di Jalan Parung Jaya No. 12 RT 04/ RW 03, Karang Tengah - Kota Tangerang 15158.

No. Ponsel : 08975480464

Email : abdurrozaq17@gmail.com

Riwayat Pendidikan : Penulis mengawali pendidikan di SD Negeri Kembarangan Selatan 01 Pagi Jakarta Barat pada tahun 2004 - 2010. Setelah itu, penulis melanjutkan ke SMP Negeri 134 Jakarta hingga tahun 2013. Kemudian kembali melanjutkan ke SMA Negeri 65 Jakarta dan lulus tahun 2015. Pada Tahun yang sama penulis melanjutkan ke Universitas Negeri Jakarta (UNJ), Program Studi Matematika, melalui jalur PENMABA. Pada pertengahan bulan Desember tahun 2019 penulis telah memperoleh gelar Sarjana Sains dari Program Studi Matematika, FMIPA, UNJ.

Riwayat Organisasi : Selama di bangku perkuliahan, penulis aktif di berbagai organisasi kemahasiswaan. Dalam tahun pertama, penulis menjadi bagian dari Departemen Infokom BEM Matematika. Pada Tahun kedua penulis menjadi bagian dari Departemen Maisyah (Kewirausahaan) Lembaga Dakwah FMIPA (Masjid Ulul Albaab).

Riwayat Pekerjaan : Penulis mulai menjadi pengajar matematika sejak tahun 2017. Pada tahun 2017, penulis menjadi pengajar privat matematika di Duta Bintaro dan Villa Meruya. Pada tahun 2018, penulis mendapat kesempatan menjalani magang di Kementerian Perdagangan RI selama 1 bulan.